

Praktikum “The Bioinformatics Lab”

Week 1: Debian Linux on a USB Stick

Stefan Seemayer

April 19th, 2010

For the first week of the “Bioinformatics Lab” Praktikum, we were assigned to install a Debian Linux server onto a bootable USB stick. The following protocol describes the steps performed.

1 Boot CD Preparation

The first step of installing a Debian 5.0 stable (“lenny”) server was to go to the Debian website (www.debian.org), choose “obtain a copy” and find the CD image suitable for the server architecture and bandwidth requirements.

1.1 CD Image Versions

Depending on whether a high-bandwidth internet connection is available during installation, either a self-contained, “large” installation CD image can be downloaded or a smaller “netinst” image can be used that only contains the bare essentials for installation and will download all additionally selected packages from the internet. Since an internet connection was available during setup and since security updates for packages will have to be downloaded anyways, I chose a < 180 MB “netinst” image.

No matter which type of installation image was selected, I recommend to always have a working wired internet connection available during the installation process as wireless support appears to be unstable and with no connection available, the network will not be automatically configured¹.

1.2 System Architecture

In order to fit the operating system to the machine it is installed on, it is furthermore important to choose an installation image matching the computer’s

¹If really necessary, Debian can be installed from a full 700 MB CD image without any internet connection. The steps needed to manually configure the connection later on can be found at <http://wiki.debian.org/NetworkConfiguration>. Additionally, software mirrors may not be set up correctly, so the mirror list might have to be edited manually. As said before, a working connection is recommended.

processor architecture. The main choices for desktop machines and laptops are the Intel 32-bit architectures (i386) and the AMD and Intel 64-bit (amd64) architectures.

Since 64-bit processors can emulate the 32-bit architecture, an i386 image can be installed on any desktop or laptop machine. My laptop is a fairly old 32-bit machine, so I chose the i386 image and burned it onto a standard 700 MB CD-R. I then re-booted the machine from the newly created CD and picked the installation option from the welcome screen.

2 Partitioning

Before anything can be installed on the USB stick, its memory needs to be prepared to store a Linux installation, for example by segmenting the volume into multiple partitions of different purposes, in our case, the `/boot` partition containing files required for booting the system such as the boot loader and kernel, the `/swap` partition which is used to store temporary data and increase system performance, and the “root” `/` partition that stores everything else. Other partitioning setups would store both `/boot` and root in one partition and only use one external partition as a swap area.

For this week’s assignment, we were given the choice of two different partitioning configurations: Using a unencrypted `/boot`, `/swap` and `/` partition, or, for the advanced users, creating an unencrypted `/boot` and having the remaining partitions on an encrypted logical volume. In this protocol, both variants are presented.

Whatever partitioning scheme is chosen, it is important to be extremely careful when operating the partitioning application as mistakes such as writing to the wrong hard disk can easily lead to destroying the host system the USB stick is connected to. I recommend against using any type of “guided” partitioning as this means giving up control over how the system is partitioned and clicking through the option screens, accidentally picking the wrong volume to work on can again lead to quick and painful data loss. For my system, the 8 GB USB stick was found under `/dev/sdb` and my laptop’s hard disk was located at `/dev/sda`.

2.1 Unencrypted partitioning

The simpler partitioning scheme of the two options is to use three unencrypted partitions - 512 MB for `/boot`, 512 MB for `/swap` and 7 GB for `/` (root). The steps required to set up such a partitioning are the following:

- Locate the USB stick (8 GB volume) in the partitions overview, select the 8 GB FAT32 partition that came with the USB stick and delete it.
- Select the free space on the USB stick and add a new partition of size 512 MB, at the beginning, using the ext3 file system. As a mount point, choose `/boot` and toggle the bootable flag to “on”.

- With the remaining free space, create another 512 MB partition and select “swap area” in the “use as:” field.
- The remaining 7 GBs of memory go to a third and final partition with the ext3 file system, mounted at `/`.
- Double-check if everything was configured correctly and select “finish” to perform the actual partitioning.

2.2 Encrypted partitions with LVM

Using encrypted partitions is a lot more complicated than a classical partitioning setup since parts of the file system are loaded from logical volumes as provided by the Logical Volume Manager (LVM). For both installation and getting the system to actually run, many more steps are required:

- Choose manual partitioning when asked how to partition.
- Delete the 8 GB FAT32 partition that originally resides on the USB stick.
- Create a 512 MB ext3 partition mounted as `/boot`, switching on the bootable flag. This will be used for the kernel and boot loader and contains the software required for decrypting the actual storage partitions.
- Create a 7.5 GB partition used as a “physical volume for encryption”.
- Choose “Configure encrypted volumes” (very top of the menu) to save the partition table and format the encrypted partition. This will also ask for an encryption passphrase. If running on a computer that contains a previously installed linux system, the unencrypted `/swap` partition belonging to this installation may have to be temporarily disabled.
- After formatting, another logical 7.5 GB volume appears in the volume list that represents the encrypted storage. It already contains an ext3 partition, but in order to add multiple partitions into a virtual volume using the Logical Volume Manager (LVM), its usage has to be changed into “physical volume for LVM”.
- On the partitioning overview, over “Configure encrypted volumes”, the new option “Configure Logical Volume Manager” becomes available. Select it to set up the logical encrypted partitions.
- First, create a volume group, selecting the encrypted partition as a physical storage (space bar to check the box) and assigning it a name (e.g. “crypt”). Then, create a 512 MB logical volume in that volume group, naming it “swap”, and another logical volume “root” taking up the rest of the space. Confirm the changes to return to the volume overview with the newly created logical volumes.
- For the “swap” partition, select “swap area” in the use as field.

- For the “root” partition, select the ext3 file system and the “/” mount point.
- Double-check if everything was configured correctly and choose “finish” to begin actual partitioning.

3 Installation

After partitioning is completed, the actual installation can take place. First of all, a “root” administrator account is created for which the password will be prompted, followed by account data for a restricted user.

3.1 User Account Creation

In all linux systems, the default super user account name is “root”. Since this user has full access to all components of the system, it is not recommended to be logged in under this account for everyday work, but only for performing administrative actions on the system. For security reasons, a very strong password should be chosen on production systems and it might be advisable to fully de-activate the root account and instead use sudo after the installation was completed.

For the regular user account, the installation program first asks for the full name and then for the logon username. I chose my last name as a username, but different policies apply to different groups.

3.2 Mirror Selection

Since I chose to install from a netinst CD image, additional software has to be downloaded from the internet. For this purpose, the Debian project provides a large list of software mirrors all over the globe. It is recommended to select one that is geographically close for higher download rates, for example the Uni Frankfurt mirror.

3.3 Additional Package Selection

With a mirror selected, the installer presents the option of selecting additional packages to be installed from the web. Since we will be configuring our system bit by bit, we’re aiming at a minimal system for now, so I de-selected all packages except for the “Standard system”, which contains command-line tools that will make interaction with the server easier later on.

3.4 Bootloader Installation

After the packages have been downloaded and installed from the mirror server, the installer finally prompts whether to install the bootloader to the master boot record. Since it would install GRUB to the computer’s hard disk instead

of the USB stick, it is important to pick “no” here and instead manually specify the path of the USB stick, which was `/dev/sdb` on my setup.

After GRUB is installed, installation is completed and it is time to boot up the server for the first time.

4 Initial boot

As nicely described in the Debian Installation Manual, booting up for the first time is a “Moment of Truth”, showing whether the setup was successful or if additional configuration steps are required.

4.1 Problems by booting from the USB stick

After re-starting the system, GRUB opened up with a choice of Debian Linux in a single- and multi user configuration, with single-user being a fallback option. Simply picking the Debian Linux entry resulted in a file not found error, since the hard disk configuration of install time is different from the hard disk configuration at run time. As mentioned in the hints on the wiki, I therefore had to edit the boot configuration, picking root `(hd0,0)` instead of root `(hd1,0)`.

This modification caused the the `initrd` image to load and it attempted to open up the encrypted partition, but the corresponding volumes could not be found. It took me some time of exploring various config files and searching the internet to find out that this was a problem caused by a delayed load of the USB stick which could be ameliorated by adding `rootdelay=10` to the `initram` script options in the GRUB boot menu. Since it is recommended to add the same line to `kopt` in the `/boot/grub/menu.lst` file for it to remain active after automatically updating the `menu.lst` file by a script, I made all changes in the `menu.lst` file to make them permanent for every boot.

4.2 Additional Packages

To make the newly installed system not only operable, but more comfortable to use, some additional packages were installed.

4.2.1 Upgrade to GRUB2

As recommended by the tips list, GRUB was then replaced by GRUB2 using the command-line version of `aptitude` by logging in as root and doing

```
$ aptitude install grub2
```

When asked whether I want to chain-load GRUB, I chose no and used

```
$ grub-install /dev/sdb
```

to install GRUB2 in the USB stick’s Master Boot Record.

4.2.2 Sudo

Since switching back and forth between the root and restricted user account is tiresome and remaining in the root account for all purposes is insecure, I decided to install sudo by running

```
$ aptitude install sudo
```

and adding the group “sudo” to the list of sudoers by doing

```
$ visudo
```

and adding the line (including the % percent sign)

```
%sudo ALL = (ALL) ALL
```

To add my restricted user name “seemayer” to the “sudo group, I finally ran:

```
$ usermod -a -G sudo seemayer
```

Any command that requires superuser privileges can now be run from “seemayer”’s account by prefixing it with sudo and re-authenticating using the restricted user’s password.

4.2.3 Lynx

To verify whether sudo was installed and configured correctly, I installed the text-only browser lynx by changing to my restricted user account (su seemayer) and running

```
sudo aptitude install lynx
```

to install the browser. This way, I can browse documentation while working on the server more conveniently.